AD-A221 387

DTIC FILE COPY

(2)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	12 Mar 90	FINAL 01 Jan 89 to 31 Dec 89

4. TITLE AND SUBTITLE

Adaptive Control of Visually Guided Grasping in Neural Networks

5. FUNDING NUMBERS

AFOSR-89-0-0030
88-NL-209
2313/A8
61102F

6. AUTHOR(S)

Dr Michael Kuperstein

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Neurogen Laboratories Inc.
325 Harvard Street, Suite 202
Brookline, MA 02146

8. PERFORMING ORGANIZATION REPORT NUMBER

AFOSR-TR- 90-0420

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

JOHN F TANGNEY
AFOSR/NL
Building 410
Bolling AFB, DC 20032

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFOSR-89-0-0030

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

We present a theory and prototype of a neural controller called INFANT that learns sensory-motor coordination from its own experience. INFANT adapts to unforseen changes in the geometry of the physical motor system and to the location, orientation, shape and size of objects. It can learn to accurately grasp an enlongated object without any information about the geometry of the physical sensory-motor system. This new neural controller relies on the self-consistency between sensory and motor signals to achieve unsupervised learning. It is designed to be generalized for coordinating any number of sensory inputs with limbs of any number of joints.

14. SUBJECT TERMS	15. NUMBER OF PAGES	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
U	U	U	U

90 05 10 119

AFOSR-TR- 90-0420

AFOSR FINAL REPORT

ADAPTIVE CONTROL OF VISUALLY GUIDED GRASPING IN NEURAL NETWORKS

Neurogen Laboratories Inc.

Project Summary

Research performed for AFOSR focused on visually guided and sequential sensory-motor coordination. It was inspired by concepts and data from both neuroscience and infant development. The research began with a prototype simulation model of the INFANT controller, which was able to learn a "sense of space" by its own experience. Progress in this research has extended the INFANT controller in two ways:

1) The simulation model was implemented with real targets and movements using two stereo TV cameras and a multijoint manipulator.

2) The theory of sensory-motor coordination was extended from single movements to movement sequences.

The neural network controller in the proposed study has a number of application benefits. The controller will deal effectively in novel working environments such as in space because of its ability to deal with unforeseen changes in the mechanical plant and actuators. Its adaptability will allow continuous self-calibration and its generic design will allow it to be implemented in many different robots. The parallel feedforward control architecture will make robot control very fast and the overlapping modifiable neural weights will allow fault tolerance. This will greatly reduce tooling costs, setup time and failure in unforeseen environments.

16 MAR 1990

Project Objectives

The proposed work had two aims:

1) To take an existing prototype neural network model of adaptive hand-eye coordination and implement it with real targets and movements using two stereo TV cameras and a multijoint manipulator.

2) To extend the current network model of eye-hand coordination so that it will control movement sequences.

Implementation of an Adaptive Hand-Eye Controller for Single Movements

Abstract: We present a theory and prototype of a neural controller called INFANT that learns sensory-motor coordination from its own experience. INFANT adapts to unforeseen changes in the geometry of the physical motor system and to the location, orientation, shape and size of objects. It can learn to accurately grasp an elongated object without any information about the geometry of the physical sensory-motor system. This new neural controller relies on the self-consistency between sensory and motor signals to achieve unsupervised learning. It is designed to be generalized for coordinating any number of sensory inputs with limbs of any number of joints. INFANT is implemented with an image processor, stereo cameras and a five degree-of freedom robot arm. Its average grasping accuracy after learning is 3% of the arm's length in position and 6 degrees in orientation.

DTIC

Keywords: Neural Networks, Adaptive Motor Control, Sensory-Motor Coordination

Introduction

The human brain develops accurate sensory-motor coordination in the face of many unforeseen changes in the dimensions of the body, strength of the muscles and placements of the sensory organs. This is accomplished for the most part without a teacher. A simple version of this skill has now been implemented in a robot control system. We present a new theory and implementation that suggest how at least one type of adaptive sensory-motor coordination might be developed and maintained by animals as well as robot controllers. The theory relies on the self-consistency between sensory and motor signals to achieve unsupervised learning. The self-consistency hypothesis is an extension of results from developmental studies in coordination behavior. Studies in the kitten (Held and Hine, 1963) show that visually guided behavior develops only when changes in visual stimulation are systematically related to self-produced movement. The hypothesis is also consistent with the motor theory of speech perception (Liberman et. al., 1967; Williams and Nottebohm, 1985).

The new theory also relies on the topography of neural units in a network. Topography is the ordered contiguous representation of inputs or outputs across a surface with possible overlap of neighboring representations. Topographic mappings have been found in most sensory and

motor brain structures (Kandel and Schwartz, 1985).

This study combines the constraints of self-consistency and topography for adaptively coordinating a multijoint arm to reach an elongated object arbitrarily positioned in space, as viewed by two cameras. The architectures of neural networks in this study consist of arrays of simple identical computational elements, called neurons, where each neuron can modify the relationship between its inputs and its output by some rule. The power of these neural networks comes from a combination of the geometry used for the connections, the operations used for the interaction between neurons and the learning rules.

Adaptive Control

We define adaptive control to mean one that can learn and maintain accurate performance even after unpredictable changes are made in either the geometrical, mechanical or sensing parameters or from partial internal damage. Learning must be achieved without a teacher. Changes in a motor plant after wear include :

1. Transformation of sensor signals to actuator movement.

2. Link lengths.

3. Signal noise.

4. Potential internal processor faults .

To be useful, an autonomous controller must operate in real time, learn and maintain its own calibration and be expandable to accommodate many robot joints. Currently, autonomous robots are controlled by either direct program control, teaching pendants, inverse kinematics or classical adaptive control techniques. None of the first three methods are adaptive. As for classical adaptive control techniques, they require a model of the robot plant and actuators, which may be difficult to obtain beforehand. Also, multijoint inverse kinematic methods are computationally intensive which slow down the control process considerably .

The specific problem that this study addresses is controlling a robot to reach objects in space. This is laid physically as a 5-degree of freedom arm being guided by stereo cameras to reach an elongated object (figure 1).

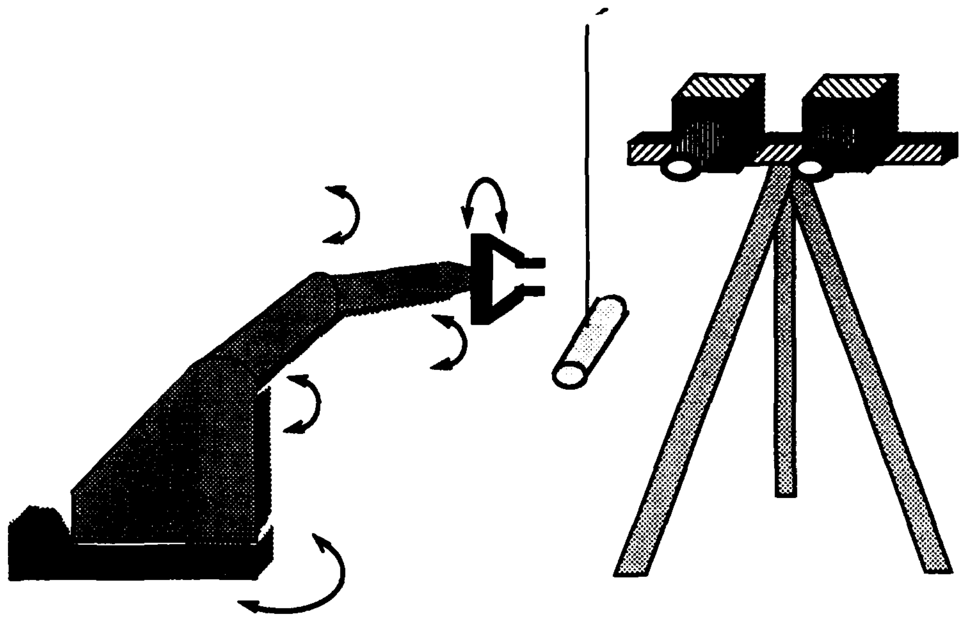

Figure 1. Schematic of the 5-degree of freedom arm and stereo camera system.

. Neural Controller Constraints

The design of the neural network that adaptively controls the motor plant has a number of constraints.

1. The Neural Controller design must not contain information about the plant or actuator characteristics. By being independent of any plant, it can be generically applied to all plants.

2. The neural network has to learn the association between the stereo image of a an object and the pattern of motor signals to all the joints.

3. The neural network computation must be parallel so that is can be generalized to many joints without increasing processing time.

The specific problem of reaching an elongated object in space has another constraint:

Since an object can be at any location in 3D space and at any orientation, the controller must have some way of processing 3D information from stereo views of the object.

These constraints were first used in a previous study to create a 3D computer simulation of a neural network controller (Kuperstein, 1987, 1988a, 1988b). The current implementation is an extension of that simulation. The Neural Controller is based on newly discovered distributive neural representations and computations. The following sections will discuss the design of the Neural Controller in increasing levels of detail.

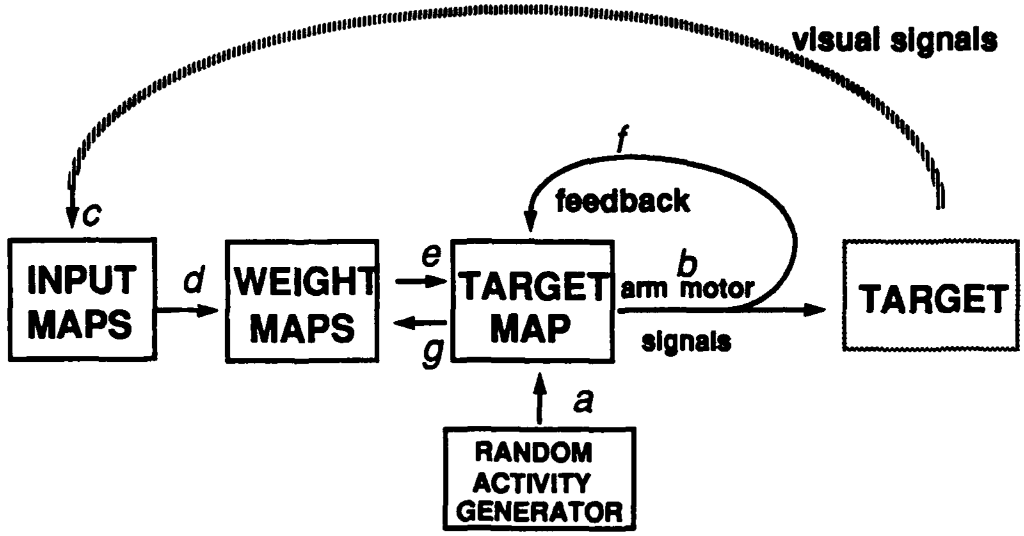

Figure 2. The circular reaction. Self-produced motor signals that manipulate an object target are correlated with target sensation signals. The sequence for training is *a, b, c, d, (e+f), g*. Correlated learning is done in step *g*. After the correlation is achieved, target sensation signals alone can evoke the associated motor signals to accurately manipulate the target. The sequence for performance is *c, d, e, b*.

Neural Controller System

The Neural Controller is designed to learn self-consistency between sensory and motor signals without supervision. In general it operates in two phases in a process called a circular reaction, shown in figure 2. The current use of circular reaction is an extension of one of J. Piaget's developmental stages (1952). In the first phase, sensory-motor relations are learned via correlations between input and output signals. In the second phase, the system uses the learned correlations to evoke the correct posture that manipulates a sensed object.

The block diagram in figure 2 shows the major modules that embody the Neural Controller During learning in the first phase, a random generator first produces random postures of the multijoint arm in space, while the gripper holds an object such as a cylinder. Then the stereo cameras snap the image of the arm holding the object. These images are transformed into neural input maps. The signals coming from the input maps are modulated by a set of weight signals to produce target signals. The weights constitute the global association between all possible images of an object and the arm motor signals.

The target map outputs are compared with the random outputs that generated the arm posture to produce error signals. These error signals are then used to update the weight signals which improves the reaching performance.

During performance in the second phase, the object is located somewhere free in space. The sequence of transformations from the stereo cameras to the target map is reiterated as in the learning phase. The resulting target map outputs then drive the arm actuators. Because of the learned associations between the object images and the arm motor signals, the hand will superimpose on the object.

The control performance process is achieved without feedback and without knowledge of the spatial relations of the mechanical system.

Self Consistent Sensory-Motor Learning

The inputs to the Neural Controller are camera images and the outputs are arm signals. Stationary stereo cameras register images composed of light intensity I_{ij}, i=1,2...I, j=1,2...J. while arm signals a_m (m=1..M) activate four joint angles of the arm (see figure 1): the base, the shoulder, the elbow, and gripper roll. The gripper is controlled by conventional methods. The Neural Controller has no a priori knowledge of spatial relations of the mechanical system.

On learning trial n, the arm signals are first randomly generated in user coordinates, which range from 0 to 1, and are transformed into joint angles for the arm. These joint angles in turn are transformed in the required stepping motor steps of each joint. In this simplest case, the joint angle of the limbs are computed to be linearly proportional to the user coordinates. However, any one of many monotonic functions of user coordinates can be chosen with similar results. Random activation of the arm joints leads to an arm posture with the two fingered hand initially holding a cylinder.

Then, the stereo cameras register two views of the arm holding the cylinder. Each image is first processed for novel contrast in the scene as compared against the background. This allows the controller to be insensitive to potentially distracting background objects. Next, the novel contrast view is processed for the cartesian location of the center of highest visual contrast or peak location. Visual contrast is determined by a combination of a high spatial contrast convolution followed by a low spatial smearing.

The peak locations identify the target to be grasped. These locations are first transformed into neural input maps. All the neural input maps have the same format. They are composed of linear arrays of discrete values across one parameter dimension.

There are six neural input maps for gaze, g_{pqn} (p=1,2; q=1,2,3; n = 1,2...N) which represent the x position and y position for each camera and the x position disparity and y position disparity for each camera. Disparity is defined as a monotonic function of the difference between the peak locations at each camera. The gaze maps are generated as unimodal distributions of activity. Any one of a large family of transformations can be chosen without affecting the results. The main criterion for these transformations is that the positions of unimodal distribution peaks vary monotonically with a parameter dimension which in this case is the Cartesian location or Cartesian disparity. For this implementation we used an inverse parabola function where x_{pq} is the gaze position or disparity parameter, n is the map

index and α and β are normalization constants.

$$g_{pqn} = \max [\, 0,\ \alpha\,(1 - (x_{pq} - n)^2/\beta)\,] \tag{1}$$

There are two benefits to choosing a unimodal distribution that results in overlapping activation of neighboring neural elements: 1. The system can effectively average out signal noise and 2. the system provides accurate interpolation for reaching objects that have never been encountered during learning.

Each camera receives a two-dimensional visual projection of the cylinder target in space. Each image V_q is processed for visual contrast orientation. There are twelve neural visual maps v_{rqn} (r=1..4; q=1,2,3; n=1,2...N) which represent 4 graded contrast orientations (0°, 45°, 90°, 135°) for the left camera, right camera and the orientation disparity between cameras. To get a visual neural map, a measure of normalized orientation, d_{rq} is first computed. This measure is achieved by convolving an image neighborhood around the center of contrast location from each camera with an orientation kernel k_r. In this convolution, q is the left or right camera and r is the orientation direction.

$$d_{rq} = V_q * k_r \tag{2}$$

The kernel matrices, k_r, have the same nonpositive coefficients everywhere except along one string in one of the four orientations, r. The coefficients in that string are all the same positive number. Other orientation-response functions can be used with similar results.

Next the measures of contrast orientations are normalized to each other, so that the orientation measures will be insensitive to different lighting conditions:

$$d'_{rq} = d_{rq} / \Sigma_{rq} (d_{rq}) \tag{3}$$

Note, that since the measure of orientation is sampled for only a small neighborhood around the contrast center of the object, this measure is insensitive to object size.

The orientation measure is then used to generate the visual neural maps (the same as the gaze maps) where δ and λ are constants.

$$v_{rqn} = \max [\, 0,\ \delta\,(1 - (d'_{rq} - n)^2/\lambda)\,] \tag{4}$$

After the neural maps are computed, they can be used to generate arm signals, a'_m through their respective weight maps. Each gaze and visual neural map is connected with a weight map for each joint, m = 1..4 . The modifiable target weights in these maps act as associative gates between sensation and posture. In this equation $w_{pqn<m>}$ are the modifiable weights from all input map elements to all joints.

$$a'_m = \Sigma_{pqn} (g_{pqn} w_{pqn<m>}) + \Sigma_{rqn} (v_{rqn} w_{rqn<m>}) \tag{5}$$

Note that weight values can be negative. All weights are initialized to 0.

The model develops self-consistency and improves its performance by modifying the target weights. The weights are changed by a learning rule during each trial, which develops the correlation between topographic sensory signals and topographic motor signals across all trials. The learning rule minimizes the difference between the actual (random) and computed motor signals. Thus, the differences or errors ε_m are

$$\varepsilon_m = a_m - a'_m \tag{6}$$

Minimizing these differences while allowing global convergence requires changing all active target weights by a small amount. In this equation k is the learning trial number and σ is the learning rate.

$$w_{(k+1)pqn<m>} = w_{(k)pqn<m>} + \sigma \varepsilon_m g_{pqn} \tag{7}$$

$$w_{(k+1)rqn<m>} = w_{(k)rqn<m>} + \sigma \varepsilon_m v_{rqn} \tag{8}$$

The learning rule states that the target weights corresponding to those sensory inputs that are active, are changed by an increment that depends on the component of an error in the respective joint. This component specific learning occurs back in the weight maps. With this incremental learning rule, the computed motor signals for all targets converge towards the actual motor signals in successive trials, thereby minimizing ε_m.

When learning has converged, accurate performance can begin. The object is located somewhere free in space. The sequence of transformations from the stereo cameras to the target map is reiterated as in the learning phase (equations 1 - 5). The resulting target map outputs a'_m then drive the arm actuators. Because of the learned associations between the object images and the arm motor signals, the hand will superimpose on the object.

During the learning phase, similar object orientations can be reached by two extremely different wrist roll angular positions. For example, a vertical orientation of a cylinder can be reached by a 0 degree wrist roll or 180 degree wrist roll. This presents a problem in trying to determine which angular wrist position should be used when the cylinder is vertical.

The approach we took was to introduce the concept of multiple virtual joints that make up a single physical joint. The purpose of virtual joints is to distinguish which one of multiple, different behaviors is most appropriate for a similar presentation of an object. The idea is that the controller first associates different behaviors with overlapping ranges of object presentation. Then for a single object presentation, the learned behaviors compete for being most appropriate.

Virtual and physical joints both have weight maps associated with them. While the weight maps for physical joints are always updated during learning, the weight maps of virtual joints are updated only for the corresponding subrange of a joint postures. For example, one virtual joint corresponds to the range of wrist roll from 0 to 90 degrees and the second virtual joint corresponds to the range of wrist roll from 90 to 180 degrees.

During performance, the neural controller makes a decision about which of the multiple virtual joints and associated weight maps are most appropriate to the orientation of the object. Neural weight maps of all virtual joints are concurrently activated by the same object. Then all virtual joints produce computed arm signals. How can the controller distinguish an appropriate arm signal from an inappropriate arm signal? It can not distinguish on the level of activity alone since appropriate and inappropriate behaviors can both generate a large or small arm signal. Our solution is to incorporate antagonistic arm signals for all virtual joints whose sum is a constant. Then appropriately learned behaviors will generate antagonistic arm signals that sum to that constant while inappropriate behaviors will generate a low level of activity smaller than the constant.

Sensory Error Learning

Additional learning can occur to fine tune and maintain accurate grasping performance in the face of unexpected changes in the geometry of the physical plant. This learning is based on errors derived from visual feedback after an attempted grasp. This sensory based scheme was rejected as a method for learning fundamental sensory-motor coordination. To learn a motor position using sensory based errors would require a priori knowledge of the correspondence between sensory errors and motor errors. Our self-consistent learning scheme avoids this requirement and thus allows motor learning from any relation between the cameras and the arm. However, once a global association between sensation and posture is learned, it can be used to define the correspondence between sensory errors and motor errors. This section describes this secondary sensory-based learning.

In an attempt to grasp the target, for example a ball, the gripper may miss the target. In this case, the mismatch between the position of the ball and the gripper is the source of an error. Our sensory based learning scheme requires the object and the gripper to be treated as two distinct sensory inputs with separate neural input map representations. These are mapped into different computed arm postures using the same current weight maps. The computed arm posture for the gripper, a''_m is determined by the following equation where g'_{pqn} is the input neural map for the gripper.

$$a''_m = \Sigma_{pqn} (g'_{pqn} w_{pqn<m>}) \quad (9)$$

The differences between the computed arm postures are the motor errors, ε'_m, used to update the current weight maps:

$$\varepsilon'_m = a'_m - a''_m \quad (10)$$

This error is used to update the neural weights similar to equations 7,8. The weights are updated in a region centered around the peak of the object's input neural map, g_{pqn}. In this equation ϕ is a learning rate.

$$w_{(k+1)pqn<m>} = w_{(k)pqn<m>} + \phi\, \varepsilon'_m\, g_{pqn} \tag{11}$$

The updated weights represent an improved correspondence between the where the object is seen with where the gripper should move. This sensory-based learning scheme can be repeated in a sequence of improved reaching performances up to a specified accuracy.

Neural Controller Implementation

The neural network controller is implemented on an industrial robot arm and commercial image processing system. The implementation of the Neural Controller was achieved by developing software that embodies the Neural Controller processing into an image processor. Equations and procedures for the INFANT were translated into a sequence of image processing commands and robot commands from both the image processing software and robot control software. During implementation it was determined that processing weight maps values required at least 16 bits of resolution in order to provide sufficient dynamic range for accurate learning. Each map has a population of 640 neural elements. The location of the peak of the inverted parabola along the neural map dimension is proportional to the size of the input parameter.

Neural Controller Performance

The Neural Controller was able to accurately reach a cylinder that was arbitrarily positioned anywhere in space within arm's reach. It could not only reach the cylinder that it was trained on, but could also accurately reach many other elongated objects including cylinders of different diameters, lengths and visual contrasts, as well as a piece of paper that was rolled into an irregular elongated form.

During one learning set, the image processor experienced a bit fault (bit 5 of 16) which affected every weight map value. Despite this extreme hardware fault, the reaching performance was not significantly affected.

Performance accuracy was measured for reaching a cylinder target. The accuracy reached an asymptote after about 1,200 learning trials. The average cartesian distance between the intended reach and the actual reach was 3 % of the length of the arm. The average angular deviation between intended gripper wrist orientation and the actual cylinder orientation was 6 degrees.

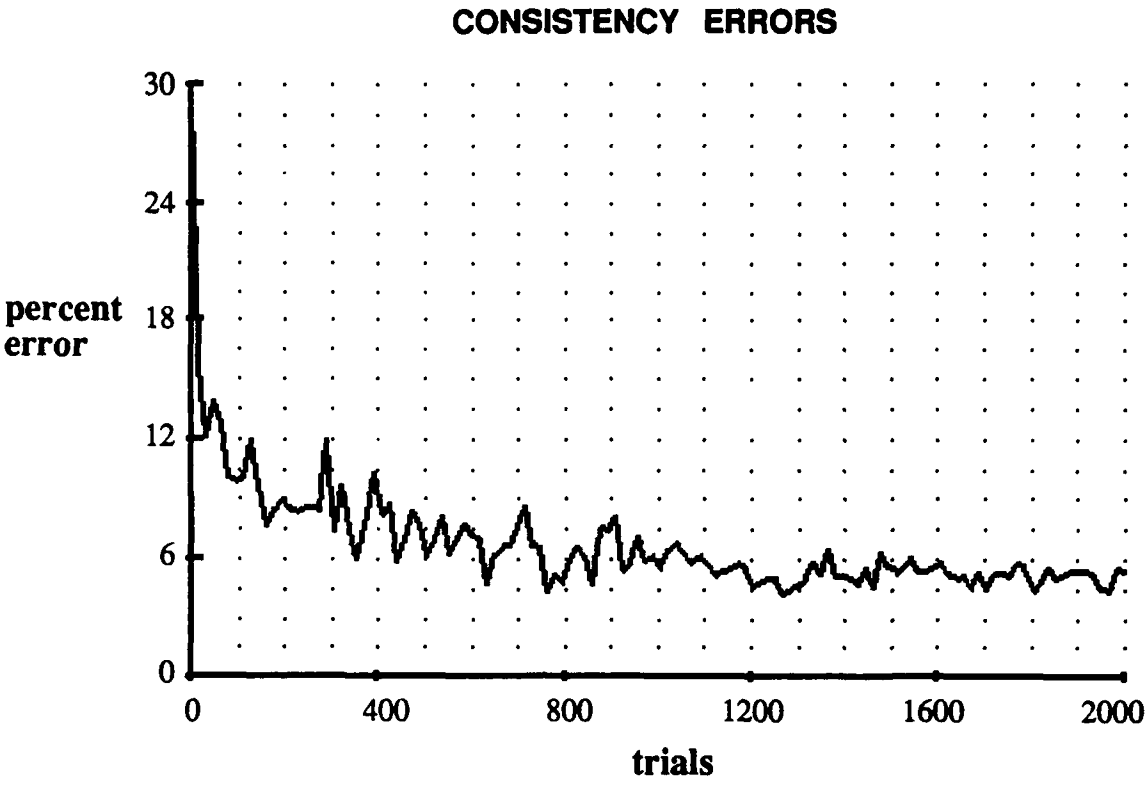

Figure 3. Converge of consistency errors over trials. The final average error is 5%.

Figure 3 shows the consistency errors across learning trials. The consistency error is determined by the following equation where m = 1,2...M.

$$(100 / M) \Sigma_m \mid \varepsilon_m \mid \tag{12}$$

Random performance would result in 50% consistency error while the final consistency error of 5% corresponds to the final average distance and orientation errors that were objectively measured.

Whole learning runs were done with different arm and camera positions and a range of neural parameters including neural map width, error rate normalization constants. The Neural Controller showed very similar error convergence for a large range of parameter choices.

Conclusions

A Neural Controller has been implemented to achieve adaptive hand-eye coordination in reaching an elongated object in space. There are four key conclusions from this study:

1. The Neural Controller learns about space by associating how the arm moves with what the cameras see. It does not use any knowledge of the geometry of the mechanical

system and there are no inverse computations nor control transfer functions.

2. Learning is achieved without any external source of error correction. This allows the controller to be completely independent of supervision. The learning for one elongated object transfers to any elongated object. Moreover, no calibration for binocular parallax is required.

3. The implementation is fault tolerant to hardware errors.

4. The Neural Controller adapts to unforeseen changes in the geometry of the motor plant.

The algorithmic flow of control is both feedforward and parallel during performance, which makes it fast. After each reach, internal feedback is used to improve subsequent performance through learning.

When implemented in custom hardware, this system will coordinate a multi-joint robot arm to adaptively reach targets in three dimensions in real time. The system will self-organize and maintain motor calibrations starting with only loosely defined relationships. The system will tolerate internal noise, partial control system damage and changes in the geometric, mechanical and sensing parameters of the robot as they occur during wear. This adaptability replaces the need for operator calibration.

The neural network controller in this study has a number of application benefits. The controller will deal effectively in novel environments because of its ability to deal with unforeseen changes in the mechanical plant and actuators. Its adaptability will allow continuous self-calibration and its generic design will allow it to be implemented in many different robots. This will greatly reduce tooling costs and setup time.

References

R. Held and A. Hine (1963) Movement-produced stimulation in the development of visually guided behavior , **J. Comp. Physiol. Psych.** 56, 872 .

E. R. Kandel and J.H. Schwartz (1985) **Principles of Neural Science** (Elsevier Science Publishing).

Kuperstein M. (1987) Adaptive Visual-Motor Coordination in Multijoint Robots using Parallel Architecture *Proc. IEEE Internat. Conf. Automat. Robotics*, March, Raleigh, N.C. 1595-1602

Kuperstein M. (1988) An adaptive neural model for mapping invariant target position. *Behav. Neurosci.* 102(1):148-162

Kuperstein M. (1988) Neural Network Model for Adaptive Hand-Eye Coordination for

Single Postures *Science*.239:1308-1311

A.M. Liberman. F.S. Cooper, D.P. Shankweiler, M. Studdert-Kennedy (1967) Perception of the speech code, **Psychol. Rev.** 74, 431.

J. Piaget (1952), **The Origins of Intelligence in Children**, translated by M.Cook, (International University Press, New York.

H. Williams, and F. Nottebohm (1985), Auditory responses in avian vocal motor neurons: A motor theory for song perception in birds. **Science** 229, 279.

Learning Goal-Directed Sequences

Problem Definition

The INFANT neural controller has been shown to accurately learn single movements to visual targets by exploring its own space (Kuperstein, 1987). This ability is accomplished by designing a neural architecture that incorporated Piaget's idea of circular reaction which he described as the first stage of infant development (1952). The primary result of implementing the circular reaction into a neural controller is the creation of self-consistency between sensation and movement. Through self-consistency, the controller develops the calibration between the topographic representation of sensation and the topographic representation of movement. To extend the capability of adaptive sensory-motor coordination toward learning sequences of goal-directed movements, a number of new developmental issues need to be addressed. These issues revolve around the structure of drive representations and the process of learning goals.

Neural Controller Design Plan

The framework for addressing these issues is based on a dialectic process of exchange between the network and its world. This process is fundamentally directed toward homeostatis or balance of the network. But far from the usual concept of homeostatis as reaching an equilibrium or asymptote of interactions, the very process of attempting to achieve a balance continually causes instabilities and reorientations. In this framework, the network is continually expanding at the limit of its comprehension of the world where its current inconsistencies create conflicts among its wants, expectations, perceptions and behaviors. As the network resolves and transcends these conflicts, its representations or maps of the world increase in complexity and scope. One might view this as a process in which the network continuously internalizes the world. By the same view, the world continuously stimulates the network to externalize itself in more complex behaviors. Like a true dialectic process, the resolution of conflicts or reorientation in homeostatis creates a new synthesis of knowledge which in turn is expressed in more complex behavior in the world. This new behavior in turn creates new unforeseen conflicts and the process continues to cycle. In essence, perceived conflicts are balanced against learned resolutions, which leads to the growth of complexity in knowledge and behavior.

According to Piaget (1954), as the mind becomes more integrated with the world by its interactions, it is also more able to distinguish itself from the world. The stages of development evolve from sensory-motor reflexes to object concept. During the course of this evolution the infant gradually shifts his representation of the world from one based entirely on his subjective actions to one based on groups and relations of objects perceived entirely independent of the self. The infant's perceptions shift from action constancies to object constancies.

The design of a neural network that can generate goal-directed movement sequences attempts to combine concepts in Freudian drive reduction, Piagetian development, operant and classical conditioning. I will first review the first Piagetian developmental stages and then describe new neural architecture concepts and their implementation and testing.

Stages in the Infant Development of Adaptive Sequence Generation

The first developmental stage of circular reaction results in calibrating self-consistency between sensation and movement through random exploration. As shown in the previous work on the INFANT controller (Kuperstein, 1988), recognition of objects in this stage is achieved only by virtue of single movement actions on an object. There is no representation of objects as permanent or as separate from actions. "In order that a recognized picture may become an object it must be dissociated from the action itself and put into a context of spatial and causal relations independent of the immediate activity." (All quotes here and hereafter are from Piaget, 1954).

In the stage of circular reactions, exploratory behavior is generated by innate reflexes, most of which relate to orientations to homeostatis or accommodation: At an autonomic level, an infant requires food, warmth, autonomic equilibrium, sleep and human contact for homeostatis. The infant can only control fulfilling these needs by crying , which is his first social means of orienting his caretaker to his needs. Crying can also be viewed as the infant's first expression of expectation. The infant expects autonomic homeostatis and cries when his system is out of balance. By sensing the consequences of crying, the infant begins the life cycle of associating expectations, consequences and behavior.

The infant also possesses some control over himself in the form of innate orienting reflexes. These include the sucking reflex, the startle reflex, rooting reflex (head turn to cheek touch), walking reflex, palmer grasping reflex, vestibular-ocular reflex, vestibular-body reflex and opto-kinetic nystagmus. Through these orienting reflexes, the infant is able to associate and calibrate topographic sensations with topographic movements. In essence, the infant orients the world to him by crying and orients himself to the world by his reflexes.

From data on brain motor physiology, we know that the mind can distinguish consequences of world movement from self-produced movement by comparing sensory inputs to recurrent motor inputs. These distinctions are the basis of creating object constancies. Kuperstein (1987) showed how a neural network could represent target position constancy using recurrent motor signals.

In the second stage, the infant's first sensory-motor expectation occurs. This is seen when a baby is distracted from his current behavior with an object and then tries to reorient back to the object. A sense of expectation is inferred by the baby's reaction of disappointment to the object's absence. The third stage of development includes an anticipation of future positions of a moving object and the representation of a whole object from partial views. During this stage, "the child no longer seeks the object only where he has recently seen it but hunts for it in a new place". In the fourth stage the " child begins to study displacements of objects by grasping them, shaking them, swinging them, hiding and finding them and thus begins to coordinate visual permanence and tactile permanence". In the fifth stage the child perceives an object as "a permanent body in motion independent of the self ...to the extent that it can be [sensed]". In the sixth stage the child represents an object as permanent in a way that is cued by sensation but maintained as mental manipulations of its predicted motion. This allows the child to look for objects behind obscuring covers.

The neural architecture presented here will incorporate aspects of the first two stages of development.

Constraints on the Architecture Design

The ultimate objective of this new neural architecture is to internalize experience into a structure of wants and expectations. This is not simply a passive selection process based on internal competition but rather an active exchange between perception and behavior. In order to implement the functions of Piaget's stages of development, the neural architecture will have to include components of circular reaction, wants, expectations, operant conditioning, learning and memory. Combining these components will allow the system to generate adaptive behavior sequences. Internalizing the interactions between inputs from stimuli and inputs from recurrent movement feedback will create an ordered map of wants and expectations. In this approach, distributed, conditioned wants and expectations give meaning to sensations and movements and eventually to the world.

From previous work, we start with the circular reaction shown in Figure 4. Its purpose is to

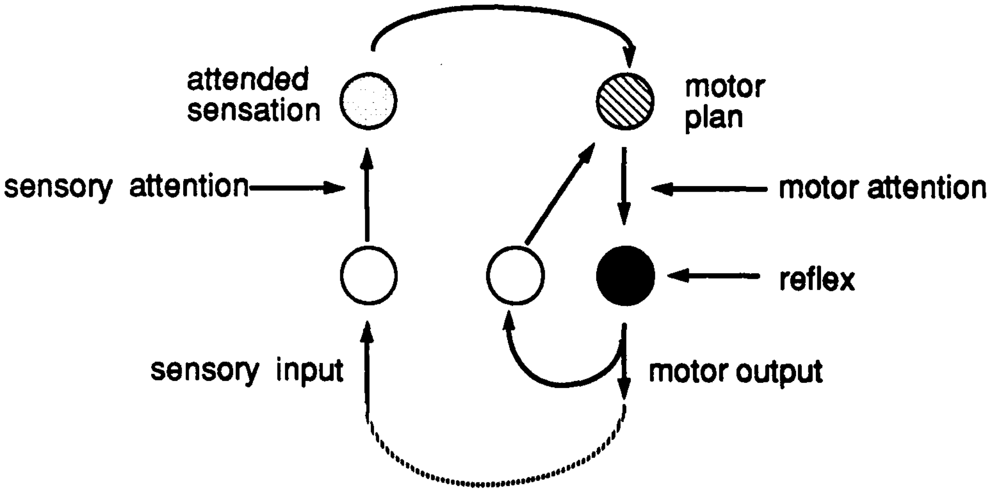

Figure 4 Circular reaction architecture

associate and calibrate sensations with movements for coordination in a number of sensory-motor systems (Kuperstein, 1988). These include coordination of hand-eye, head-eye, body-eye, head-ear and body-ear. Attention plays a peripheral role in this primitive circular reaction as a filter for relevant stimuli. Attention gates both wanted sensory inputs and motor plans.

What drives this system to behave? A reflexive or passive adaptive network can tune itself to world cues but can not grow in complexity. Its circuits can be selected by competitive responses to stimuli but it can not create new representation types. I hypothesize that **wants** drive a network to grow. Furthermore, wants are distributed in a structure that corresponds to each sensation and movement individually. **One of the novel features of this architecture is that all the required wants already exist at the beginning of the system's operation, although their connections are uncommitted.** The basic neural circuit for applying wants to behavior should satisfy the following constraints:

1. A want should enhance the sensitivity of the wanted stimulus. Among all possible stimuli, only a fraction can be attended at one time. Thus some mechanism must exist to distinguish between relevant and non-relevant stimuli. I suggest that the source of this mechanism is a distribution of wants.

2. A want should be turned off after the corresponding attended stimulus occurs. Without this negative feedback the network would continuously perseverate.

3. Any stimulus can be the source of an unforeseen distraction which may be important to the network's ability to reorient its behavior or resolve conflicts. Therefore, the ability to detect stimuli should not be modified by any network source including wants.

4. When a want is turned off, it activates a transient satisfaction signal that acts as a rewarding signal for conditioning network connections

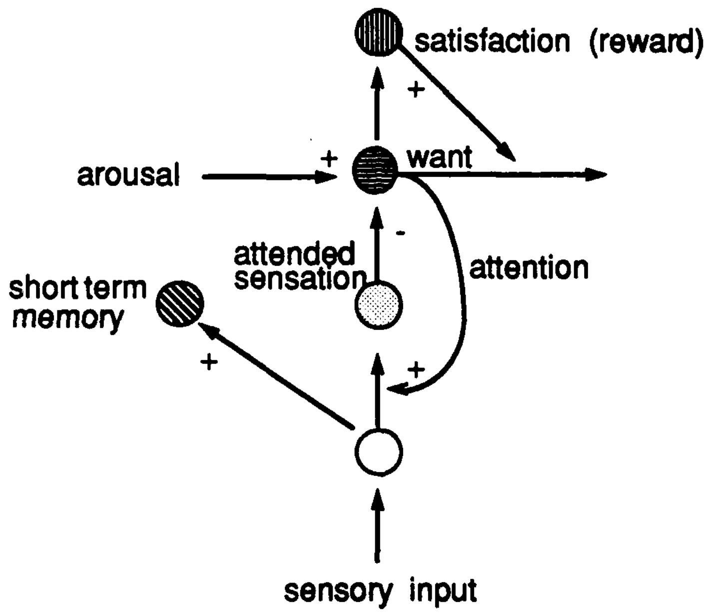

Figure 5. Drive architecture

Figure 5 shows the simplest circuit that achieves these constraints for drive. In this circuit, a stimulus is first detected. If its corresponding want is on, the sensory signal is enhanced by through a presynaptic gating mechanism and is thereby attended. The attended stimulus then turns off the want. This satisfaction event becomes a transient reward signal.

In associative conditioning, a prior behavior is enhanced by to some rewarding consequence. This requires an explicit signal of a reward, a consequence and a previous behavior. To complete the requirements of the circuit for associative conditioning, a node is added for preserving a stimulus representations for some time.

Both the circuits for circular reaction and drive can be combined to form an architecture that

can generate goal-oriented sequential movements shown in Figure 6. . This architecture is

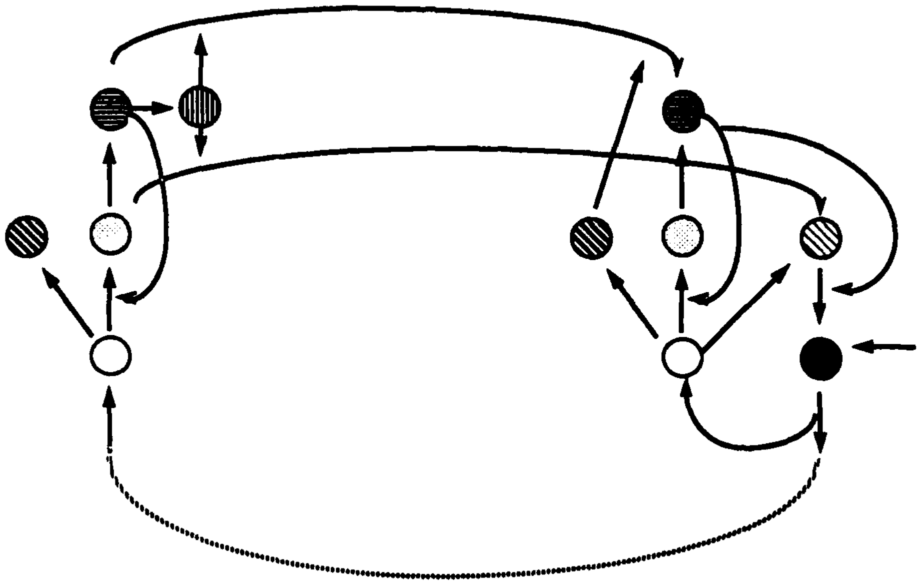

Figure 6 Architecture for Adaptive Sequence Generation:
INFANT II

called INFANT II. The distribution of wants become activated by the system's interaction with the world through associative conditioning. Associative conditioning is used to modify connections between wants that correspond to both stimuli and movements. The conditioning rule for this architecture uses the offset of a want as a rewarding event. This event conditions all wants that correspond to recently activated stimuli and movements through a Hebbian type learning rule. The effect of this rule is to set up a chain reaction among conditioned wants.

As an example lets consider a Pavlovian case. Suppose the system wants food. A bell rings and the sound is stored in short term memory. Later food appears and it is eaten. This causes the want for food to be satisfied or turned off. This rewarding event enhances the connection between the want for food and the previous sound of the bell. Consequently, when the system again wants food, it triggers the want for the bell.

In INFANT II, both the connections between wants and connections between sensory-motor representations are modified. These modified connections associate the current rewarding stimulus with a previous behavioral consequence or other stimulus. As a result, a learned behavior is activated only if the network wants to behave and the prerequisite stimulus exists. By requiring both conditions, the network can avoid reacting to spurious stimuli when they are not wanted and can avoid a directed behavior simply to wanting a behavior without a prerequisite stimulus. In essence, **future wants are created for past behaviors that lead to present rewards.**

Note that this architecture is a major extension of operant conditioning. Operant conditioning views a reward as any consequence that causes an animal to associate a stimulus with a

response in some reward schedule. What drives the animal to act is not defined. This architecture attempts to explain an animal's drive as wants from not only the autonomic system but also from distributed wants across both the sensory and motor systems.

Sequential Task Design

To test this new architecture, I have designed a simple task of learning a sequence of movements based on a set of world cues. This task is similar to one used to teach pigeons to perform tricks using operant conditioning training.

Suppose the network possesses the ability to perform four different behaviors: eat, move forward, move left, and move right. Suppose further that it can sense four different sensations: food, red light, blue light and bell. The sequence of behaviors that will be learned is determined by consequent rewards arising from exploratory behavior. I assume that the network is hungry (wants food) and therefore food is an unconditioned reward. The goal of the task is to perform the appropriate sequence of behaviors according to the sequence of world cues that ultimately lead to food. An operant conditioning training procedure is used on the network.

Although this task is simple, it highlights the potential to learn any goal oriented sequence based on world cues.

Computer Simulation

The architecture used to implement the sequence task, INFANT II, is shown in Figure 7.

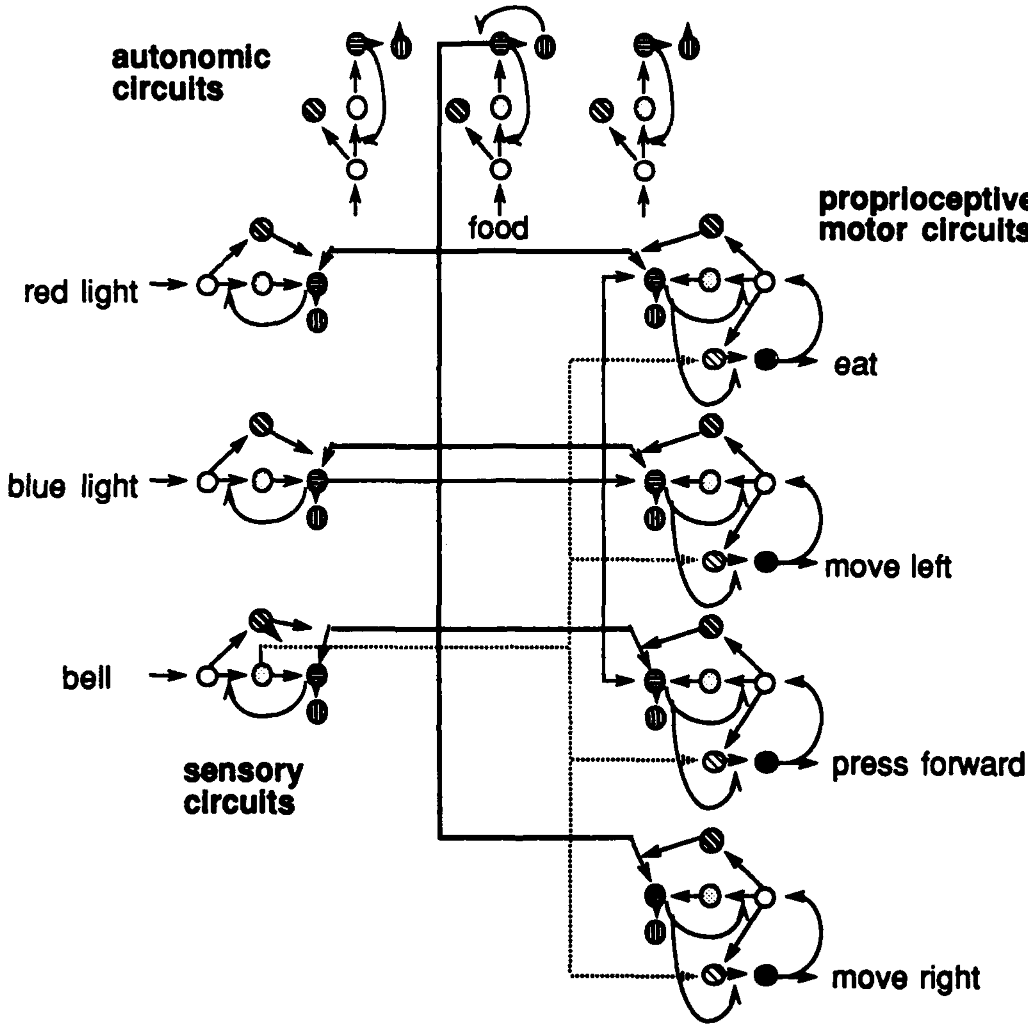

Figure 7 Architecture for Adaptive Sequence Generation: INFANT II

Only one of many connections between local circuits is shown for this fully connected architecture. In the full architecture, each want is connected to all other wants and each attended sensation is connected to all motor plans.

Modifications of connection strengths can be achieved by a number of learning rules. Two Hebbian type learning rules are used in the simulation. For conditioning the wants, the first rule states that if a satisfaction signal is on and a short term memory signal is on, the want connection at that node is enhanced. If a satisfaction signal is on and a short term memory signal is off, the want connection at that node is diminished. For sensory-motor conditioning,

the second rules states that if a sensory short term memory signal is on and a recurrent motor feedback is on, then the corresponding sensory-motor connection is enhanced. If a sensory short term memory signal is on and a recurrent motor feedback is off, then the corresponding sensory-motor connection is diminished. If sensory short term memory signal is off, then there is no change in the corresponding connection.

Initially, the network is made to want food. A typical trial consists of the following steps:

1. A stimulus is presented to the network.
2. The stimulus is detected by a sensory receptor and stored in short term memory.
3. The stimulus may satisfy a want: If the stimulus is wanted, its representation is attended which then turns off its want and a satisfaction signal is transiently activated.
4. A satisfied want conditions the learning of other wants: The satisfaction signal enhances the connection between the current want and all other wants that correspond to an active short term memory representation.
5. Sensory-motor conditioning may occur: The connection between a previous stimulus representation and all motor representations which correspond to an active proprioceptive representations are enhanced. Note that sensory-motor conditioning and want conditioning occur independent of each other in time.
6. Learned wants turn each other on in sequence: Each active want activates its enhanced connections to other wants that have been learned.
7. Learned wants allow potential actions to be executed: Active conditioned wants enable active learned sensory-motor connections to be expressed as behavior.
8. When nothing makes sense, try something new: If an unexpected stimulus or no stimulus is detected, a random exploratory behavior is generated.

The network was trained with operant conditioning techniques. The network starts out with a want for food and right light on. The network attempts exploratory behavior until it "moves-right". Food then appears, the network "eats" the food and a satisfaction signal is broadcast. Through the two learning rules, the connection between red light and move-right is enhanced and the connections from want food and the wants for red light and move-right are enhanced. Next time, the network wants food, it also turns on the want for move-right. But now the bell rings. Again the network generates exploratory movements until it "presses-forward". This results in the red light turning on which activates "move-right", which results in food.

In this chain like fashion, the computer simulation successfully learned to generate the sequential behaviors of move-right, move-forward, move-left and eat when cued by the appropriate stimuli. The following sequence of stimuli occurred only after each appropriate behavior was generated: red light, bell, blue light and food. The whole sequence is:

stimulus	**response**
1. red light	move-right
2. bell	press-forward
3. blue light	move-left
4. food	eat

There are a number of properties of this architecture:

1. It can learn as many steps as there are local circuits. The current architecture can only learn one arbitrary sequence. However, simply adding another network layer of n circuits gives n sequences.

2. The network can learn to generate different behaviors from identical stimuli, because behavior is gated by the current active want.

3. The network can be taught sequences by the standard techniques of operant conditioning. The functionality of the network fits into a new general theory of behavior development that is consistent with principles of child development.

4. The architecture does not merely learn by passive selection from frequent stimuli, but requires an active interaction between directed exploration and world consequences.

Sequence Change and Frustration

When a learned behavioral sequence changes its dependencies to a different sequence of world cues, the network must be able to adapt. To implement this property, competing wants must be able to choose an alternate sequence. Suppose want A activates want B and want B enables behavior B. From prior experience, consequence A should appear. But now instead, stimulus C appears. The network must first realize that stimulus A did not appear. This increases a frustration measure which first causes behavior B to be repeated and then causes another exploratory behavior to be generated. When the consequence of one of the exploratory behaviors, say D, results in stimulus A, and want A is satisfied, then the satisfaction signal A decreases all active prior wants such as B and conditions the new want D associated with the behavior D that lead to stimulus A. Thus the network can adaptively accommodate to changing dependencies of world cues.

Generalization Versus Differentiation

One of the main issues in the design of the neural architecture is the trade-off between the ability of the network to generalize and its ability to differentiate. A topographic network allows excellent generalizations to nearby or similar representations of stimuli or movements. This is made possible by the overlapping connections of neighboring areas of the architecture and the resulting interpolations during learning. With this architecture, how can the network possess the maximum capacity for different and unique associations? The potential for all associations must already exist before the network begins to learn. However, simply connecting all pre-existing representations to each other is very limiting. It may allow associations between all predefined representations but it does not allow for any hierarchy of attribution or scaling or chunking for stimuli or behaviors. To generate adaptive sequential behavior, an architecture must allow a number of stimuli to activate the same behavior, each at different times. But that behavior may have a different purpose for each activation. The purpose is derived from a group of active want distributions prior to the behavior.

This data flow is just the inverse of the perceptual process. There, a single sensory input can be part of many different classifications based on the context of many other sensory inputs and expectations. What type of architecture would result if the schemes for spatial processing of perception were applied to the temporal processing of behavior? The next step in generalizing this architecture is to marry the benefits of hierarchical processing with topographic connectivity.

Comparison to Previous Models

The concept of expectation is treated very differently here from the work on Adaptive Resonance Theory (ART) by Grossberg and Carpenter (1987). In ART, expectation is used as a source of pattern transformations to match incoming coded patterns. A match amplifies the coded pattern in a network, which is called resonance, and thereby indicates that the expectation correctly represents the stimuli. A mismatch diminishes the coded pattern in a network which disinhibits arousal that suppresses the most active source of expectancy and allows a new expectancy pattern to grow. This cycle iterates until the transformation of the expectancy resonates with the coded pattern.

There are a number of differences between INFANT II and ART. First, ART operates in a behavioral vacuum, while behavior is an essential component of INFANT II. Secondly, ART perseverates when an expectancy is matched as long as the stimulus is left on. INFANT II is driven to change. When a coded signal matches a want in INFANT II the signal is enhanced temporarily but then turns the want off while a satisfaction signal is transiently activated. In the ART architecture, the network could be made to change by adding habituation to the coded patterns. Third, the cycle of mismatches until resonance, attempts to mimic the concept of frustration leading to changing expectations. But how can one create new expectations without experiencing the causal relationship between behaviors and consequences? Of course alternate expectations can compete to determine which are more consistent with the current stimulus, but choosing from prior alternate expectations should not be confused with forming new expectations. While in INFANT II, frustration would occur when a behavior that was driven by a learned want did not result in the expected stimulus. Just like in ART, frustration would lead to the suppression of the most active want that was previously learned. But unlike ART, when no want matches, INFANT II would generate exploratory behavior. In essence, for ART, world cues passively select expectations through internal competition, while in INFANT II wants are learned from the interaction between behaviors and consequences.

There are also a number of differences between INFANT II and the sequence generating network of Michael Jordan (1989a, 1989b) called the forward model. His networks contain four different types of units: " The plan units are the inputs of the network....The state units receive recurrent connections from within the network and have a dual role: They provide the internal state of the system, thereby allowing the system to autonomously generate sequences of actions and they estimate the state of the environment, thereby giving the network the capability to control the environmental dynamics...The articulatory units are the outputs of the network...The task units..are the network's internal estimate of the task space."

The forward model requires as many plan units as the number of sequences and as many task units as the number of steps in a sequence. Both the forward model and INFANT II represent sequences and sequence steps by units. Plan units in the forward model have some similarities to wants in INFANT II in the sense of providing goals. However, the forward model is driven by a cost function which includes final position errors, while INFANT II is driven to satisfy wants based on temporal associations of stimuli or behavioral consequences. There is no a priori knowledge about the task implementation. INFANT II learns the task implementation based on behavioral interactions with its environment. Each world that it is exposed to, may require a different task implementation to achieve the same goals. In the forward model the task implementation is preprogrammed.

Conclusions and Applications

I have designed a new general theory for adaptive, goal-oriented sequence generation called INFANT II. It conforms to the principles of infant development and operant conditioning. I have presented the constraints by which predictive knowledge is represented by a distribution of wants and sensory-motor expectations. This knowledge is internalized through behavioral interactions with the world. INFANT II has been simulated to learn a behavioral sequence that depends on a sequence of world cues which ultimately lead to "food" for a "hungry" network. Future work will focus on expanding the architecture into a hiearchy of topographic layers as well as extending the architecture to include Piagetian developmental stages three through six. These stages will bring true adaptive goal-directed sequencing to neural networks.

INFANT II may be used to guide physiological experiments that can validate or falsify the theory and thereby gain more understanding about neural motor planning. The main engineering applications of INFANT II include adaptive navigation in novel terrains. Applications are particularly appropriate for automated transport planning for factories, hospitals, homes and in space.

References

Carpenter, G.A. & Grossberg, S. (1987) ART 2: Self-organization of stable category recognition codes for analog input patterns. **Applied Optics, 26,** 4919-4930.

Jordan, M.I.(1989a) Generic Constraints on Underspecified Target Trajectories, **Inernational Joint Conference on Neural Networks,** Washington, D.C. June

Jordan, M.I. (1989b) Action. In M.I. Posner (Ed.) **Foundations of Cognitive Science.** Cambridge. Ma: MIT Press

Kuperstein M. (1988) Neural Network Model of Adaptive Hand-Eye Coordination for Single Postures **Science.**239:1308-1311

Kuperstein M. and J. Rubinstein (1989) Implementation of an Adaptive Neural Controller for Sensory-Motor Coordination, **IEEE Control Systems Magazine.**V9:3 p.25-30

Piaget, J. (1952), **The Origins of Intelligence in Children,** translated by M.Cook, (International University Press, New York.

Piaget, J. (1954) **The Construction of Reality in the Child**, Translated by M. Cook, Ballentine Books, New York